
„MANAGER POȘTĂ FIZICĂ”

Documentație atestat – Pricop Laurențiu-Constantin
Liceul Teoretic „Petru Maior” Ocna Mureș
Profil real, specializarea matematică-informatică
An școlar 2020 – 2021

Aplicația este accesibilă live la adresa:

<https://pmail.laurcons.ro>

unde puteți folosi contul de test

baduecelmaitare

cu parola 1234

Descriere generală

Sunt o persoană căreia îi place să trimită și să primească scrisori prin Poșta Română, și de asemenea să le țin evidența. Într-o primă fază, am folosit aplicația Notițe a telefonului, apoi am trecut la un tabel Excel, iar de data aceasta am construit o aplicație web care să mă ajute să țin evidența scrisorilor care pleacă și care vin de la mine de acasă.

Structura datelor

Programul are la bază noțiunea de „destinatar” și de „scrisoare”. Acestor două tipuri de date le corespund următoarele proprietăți:

Tipul de date	Numele proprietății	Descriere
Destinatar	Numele	Numele destinatarului
	Adresa	Adresa poștală a destinatarului: strada, numărul, blocul, scara, apartamentul, localitatea, comuna, județul, codul poștal, acolo unde se aplică
	Observații	Informații care nu se aplică celorlalte câmpuri
Scrisoare	Tip	Tipul scrisorii: trimisă sau primită
	Destinatar	Destinatarul sau expeditorul scrisorii, în funcție de tipul acesteia. Termenul folosit este „destinatar” pentru ambele cazuri.
	Cod	Codul unic al scrisorii, atribuit de utilizator. Modul în care scrisorile primesc coduri se află la latitudinea utilizatorului.
	De mână?	Dacă scrisoarea a fost scrisă de mână. Se completează cu „da” sau „nu”.
	Scris	Data la care scrisoarea a fost scrisă. Se aplică numai scrisorilor trimise.
	Trimis	Data la care scrisoarea a fost trimisă. Se aplică numai scrisorilor trimise.
	Primit	Data la care scrisoarea a fost primită.
	Preț	Prețul scrisorii. Se aplică numai scrisorilor trimise.
Observații	Informații care nu se aplică celorlalte câmpuri.	

Validarea datelor

Programul adoptă o poziție iertătoare în ceea ce privește validarea acestor câmpuri: acolo unde se cer date calendaristice, sau „da” sau „nu”, programul acceptă orice secvență de caractere. Aceasta este o decizie de design, iar motivul este că există situații în care tipul datelor nu poate fi impus utilizatorului (de exemplu, în situația în care data calendaristică a primirii unei scrisori este necunoscută, dar a fost primită în mod sigur, utilizatorul poate completa cu „da” în locul unei date calendaristice corectă sintactic).

Singurele câmpuri obligatorii sunt tipul și destinatarul fiecărei scrisori, iar programul se va asigura că aceste date sunt completate înainte de a genera o nouă scrisoare.

Interfața utilizator

S-a dorit crearea unei interfețe de utilizator relativ ușor de folosit, și cât se poate de intuitivă, dar și complet funcțională, și care să minimizeze greșelile făcute de utilizator.

Manager Poștă Fizică

General Destinatari Scrisori

Listele cu destinatari și scrisori sunt accesibile prin apăsarea butoanelor din partea de sus a paginii, afișate în imaginea alăturată.

Afișarea datelor se face sub formă tabelară, paginată cu un număr de 10 intrări pe o pagină. Adresa va fi procesată și afișată într-un mod clar, concis, și exact cum se cere a fi scrisă pe plicurile trimise prin poștă.

Adăugarea de noi intrări se face apăsând butonul „Adaugă destinatar” sau „Adaugă scrisoare”, iar modificarea intrărilor se face apăsând pe ele. În momentul în care o intrare este apăsată, o fereastră se va deschide în partea dreaptă, care permite modificarea datelor. Acestea nu vor fi salvate automat. De asemenea, programul încearcă să atenționeze utilizatorul în momentul în care este pe cale de a pierde modificări nesalvate. Ștergerea intrărilor se face din aceeași fereastră laterală.

Manager Poștă Fizică

General Destinatari Scrisori

Toate Grupate

Adaugă scrisoare 1 2 3 4 5 6

Numai primite Numai trimise Toți destinatarii

În mod adițional, pagina cu scrisori conține opțiuni de filtrare, și un mod în plus de vizualizare a datelor.

Filtrarea se poate face după tipul scrisorii, sau după destinatarul acesteia.

Vizualizarea „Toate” a scrisorilor este una tabelară, similară în modul de funcționare cu cea a destinatarilor.

Vizualizarea „Grupate” a scrisorilor va lua fiecare destinatar și îi va asocia fiecăruia scrisorile primite/trimise, într-un mod intuitiv.

16 5 Ștefana 9 scrisori Ascunde

Primate				Trimise			
Cod	Primită la	De mână?	Observații	Cod	Traseu	De mână?	Observații
1 47 i4/STD	da		cod vechi iSTED/4/29.07.2020	1 41 5/STD	Scrisă: 2020 Trimisă: da Primită: da	da	cod vechi 5/STED
2 46 i3/STD	2020		cod vechi iSTED/3/26.06.2020	2 40 4/STD	Scrisă: 2020 Trimisă: da Primită: da	da	cod vechi 4/STED
3 43 i2/STD	2019		cod vechi iSTED/2/16.11.2019	3 39 3/STD	Scrisă: 2019 Trimisă: da Primită: da	da	cod vechi 3/STED
4 42 i1/STD	2019		coduri vechi i1 per 1.11.2019; iSTED/1/1.11.2019	4 38 2/STD	Scrisă: 2019 Trimisă: 2019 Primită: 2019	da	cod vechi 1/2019
				5 37 1/STD	Scrisă: 2019 Trimisă: 2019 Primită: 2019	da	cod vechi 1/2019

Detalii de arhitectură

Aplicația este împărțită în două porțiuni: partea de backend, și partea frontend. Partea backend este scrisă în întregime în PHP, și folosește Composer (un manager pentru pachete și librării PHP) pentru a integra în aplicații un număr limitat de librării PHP. Baza de date folosită este una bazată pe fișiere JSON, care nu depinde de o schemă predefinită (tabele). Partea frontend este o aplicație „într-o singură pagină” (Single Page Application) scrisă în întregime în JavaScript, folosind React.js ca framework vizual.

Backend

Aceasta este partea care relaționează cu baza de date, și care efectuează logica programului.

Baza de date este o colecție de fișiere de tip JSON (JavaScript Object Notation), organizate în folderul /db al aplicației. Librăria PHP numită SleekDB este folosită pentru interfațarea cu baza de date. Fișierelor JSON din baza de date nu li se impune o structură tabelară sau relațională, așa că programul este responsabil de a monitoriza cheile străine, sau acțiunile de modificare și ștergere. Avantajul este simplitatea sporită, și posibilitatea transferului întregii baze de date fără fișiere SQL ce trebuie rulate în phpMyAdmin. SleekDB nu este optimizat pentru viteza de accesare/indexare, din acest punct de vedere fiind mai slab decât MySQL, însă pentru o asemenea aplicație este mai mult decât perfect.

Programul expune un API (Application Programming Interface, interfață de programare a aplicației), la adresa /api, unde frontendul poate face apeluri și obține datele necesare afișării utilizatorului (de ex. lista de scrisori) sau cere modificarea datelor deja existente. Backendul este complet responsabil de validarea datelor primite prin acest API, întrucât aceste date pot proveni „de oriunde”.

Utilizatori

Aplicația implementează un sistem multi-utilizator, bazat pe username și parolă. La deschiderea aplicației, se cere autentificarea folosind datele utilizatorului, și se oferă posibilitatea înregistrării. Fiecare scrisoare și destinatar este privat, iar accesul la ele este permis numai utilizatorului de care aparțin.

Frontend

Partea de frontend este o aplicație într-o singură pagină scrisă în limbajul de programare TypeScript, folosind framework-ul React. Codul sursă al acesteia se găsește în folderul /frontend al aplicației, iar versiunea compilată în /public/app.

React se ocupă de o sumedenie de detalii vizuale și de funcționalitate, și reprezintă baza care face din această aplicație o aplicație într-o singură pagină.

TypeScript este o versiune îmbunătățită a JavaScript, care conferă acestuia din urmă un pas de compilare, pas în care limbajul poate verifica tipurile datelor folosite de aplicație. Acest pas de compilare conferă aplicației posibilitatea de a detecta erorile de tipuri de date din momentul compilării, spre deosebire de JavaScript care detectează erorile de tipuri de date numai odată ce codul este rulat, iar această funcționalitate s-a dovedit a fi foarte utilă.

Frontendul folosește o sintaxă JavaScript specială în care cod HTML poate fi imbricat direct în cod JavaScript. Această sintaxă se numește JSX și este foarte utilă în contextul dezvoltării unei aplicații bazată pe React.

```
9  ReactDOM.render(  
10  <React.StrictMode>  
11  |   <App />  
12  | </React.StrictMode>,  
13  |   document.getElementById('root')  
14  | );
```

În această poză, liniile 10-12 reprezintă cod HTML, care este tratat de JSX ca o expresie, și trimisă ca parametru funcției ReactDOM.render.

Codul HTML scris în JSX este validat în mod strict, și greșelile care sunt permise în HTML-ul obișnuit sunt strict interzise în JSX. Spre exemplu, fiecare tag trebuie închis, iar cele două moduri de închidere a tagurilor sunt prezente în poza de mai sus. Omiterea închiderii unui tag va rezulta într-o eroare de compilare din partea React.

Frontendul comunică direct cu backendul prin intermediul API-ului.

Design și iconițe

Frontendul folosește pentru design framework-ul CSS „Bootstrap”, care oferă o bază solidă pentru construcția oricărui fel de pagină web. În speță, au fost folosite două teme care fac parte din pachetul Bootswatch, numite United (tema luminoasă) și Superhero (tema întunecată).

Trecerea de la tema întunecată la tema luminoasă se poate face apăsând butonul „Mod luminos”/„Mod întunecat” din partea din dreapta-sus a aplicației. De asemenea, trecerea se face automat în funcție de ora din zi la care aplicația este pornită, fiind întunecată noaptea și luminoasă ziua.

Iconițele folosite în cadrul aplicației aparțin pachetului de iconițe al Bootstrap, numit Bootstrap Icons.

Numerotarea intrărilor

În dreptul fiecărei intrări în cadrul aplicației sunt afișate două numere, unul pe fundal roșu, și celălalt pe fundal albastru. Acestea reprezintă numărul de criteriu al intrării din tabelul curent, respectiv id-ul atribuit de aplicație intrării în cadrul bazei de date.